FS1000 系列千兆以太网卡
用户手册



FS1000 系列千兆以太网卡用户手册

资料编号 DOC-GEPCI-UM

产品版本 V1.0

资料状态 发行

版权声明

免责声明

本手册依据现有信息制作，其内容如有更改，恕不另行通知。武汉飞思科技有限公司在编写该手册的时候已尽最大努力保证其内容准确可靠，但武汉飞思科技有限公司不对本手册中的遗漏、不准确或错误导致的损失和损害承担责任。

手册使用说明

读者对象

本手册的读者对象为安装 FS1000 系列千兆以太网卡的工程技术人员和采用 FS1000 系列千兆以太网卡的软件开发人员。

内容介绍

本手册详细介绍了 FS1000 系列千兆以太网卡的安装方法，调试命令以及编程接口。

《FS1000 系列千兆过滤转发卡用户手册》共分为四章:

第 1 章 产品综述 详细阐述了 FS1000 系列千兆以太网卡的特性和产品规格。

第 2 章 产品硬件结构 详细阐述了 FS1000 系列千兆以太网卡的硬件结构和选配部件介绍。

第 3 章 驱动安装 详细阐述了 FS1000 系列千兆以太网卡在 Windows 和 Linux 下驱动的安装方法。

第 4 章 软件编程接口 详细阐述了 FS1000 系列千兆以太网卡提供的软件编程接口。

获取技术支援

客户在产品使用及网络运行过程中遇到问题时请随时与武汉飞思科技有限公司的服务支持热线联系。此外，客户还可通过武汉飞思科技有限公司网站及时了解最新产品动态，以及下载需要的技术文档。

电话：+86（027）67845143

传真：+86（027）67845062

Email: support@fisee.com.cn

网站：www.fisee.com.cn

目 录

1 产品综述

根据支持的 PCI 接口不同，FS1000 系列千兆以太网卡分为 FS1000-X 和 FS1000-E 两类，其中 FS1000-X 支持 PCI-X 规范，FS1000-E 支持 PCI-E 规范。

1.1 FS1000-X 千兆以太网卡简介

FS1000-X 系列千兆卡是一款高性能的支持 PCI-X 规范的千兆网络接入卡，可应用于 1U 或者 2U 的服务器中。该系列千兆卡可应用于大部分操作系统，符合 PCI-X1.0a 和 PCI 2.3 规范，最高可支持速率 133M 的 64 位 PCI-X。采用集成的千兆以太网控制器芯片，能够在提高性能的同时，显著降低 CPU 占用率和功耗。根据接入的千兆光口数量，该系列千兆卡分为 FS1001-X 和 FS1002-X 两种：其中 FS1001-X 提供 1 个 SFF 千兆光口，FS1002-X 提供 2 个 SFF 千兆光口。FS1000-X 千兆以太网卡主要具有以下特点:

- 提供 1 个或者 2 个千兆以太网 SFF 接口。
- 符合 PCI-X1.0a 和 PCI2.3 规范。
- 符合 IEEE-802.3ab 标准
- 支持 WFM，RIS 和 SNMP/DMI 等远程管理方式。
- 支持 windows，linux，solaris 等大多数操作系统。

1.2 FS1000-E 千兆以太网卡简介

FS1000-E 系列千兆卡是一款高性能的支持 PCI-E 规范的千兆网络接入卡，可应用于 1U 或者 2U 的服务器中。该系列千兆卡可应用于大部分操作系统，符合 PCI-E 1.0a 规范，工作在 x4 模式。采用集成的千兆以太网控制器芯片，能够在提高性能的同时，显著降低 CPU 占用率和功耗。根据接入的千兆光口数量，该系列千兆卡分为 FS1001-E 和 FS1002-E 两种：其中 FS1001-E 提供 1 个 SFF 千兆光口，FS1002-E 提供 2 个 SFF 千兆光口。

- 提供 1 个或者 2 个千兆以太网 SFF 接口。
- 符合 PCI-E 1.0a 规范。
- 符合 IEEE-802.3ab 标准。
- 支持在所有的 PCI Express 服务器插槽上使用。
- 支持 WFM，RIS 和 SNMP/DMI 等远程管理方式。
- 支持 windows，linux，solaris 等大多数操作系统。
- 与操作系统配合使用时，能够平衡多个 CPU 内核间的网络负载，从而提高多处理器系统的性能。

2 产品硬件结构

下面以 FS1002-X 为例，介绍 FS1000 系列千兆以太网卡的硬件结构特点。

2.1 FS1002-X

2.1.1 硬件结构

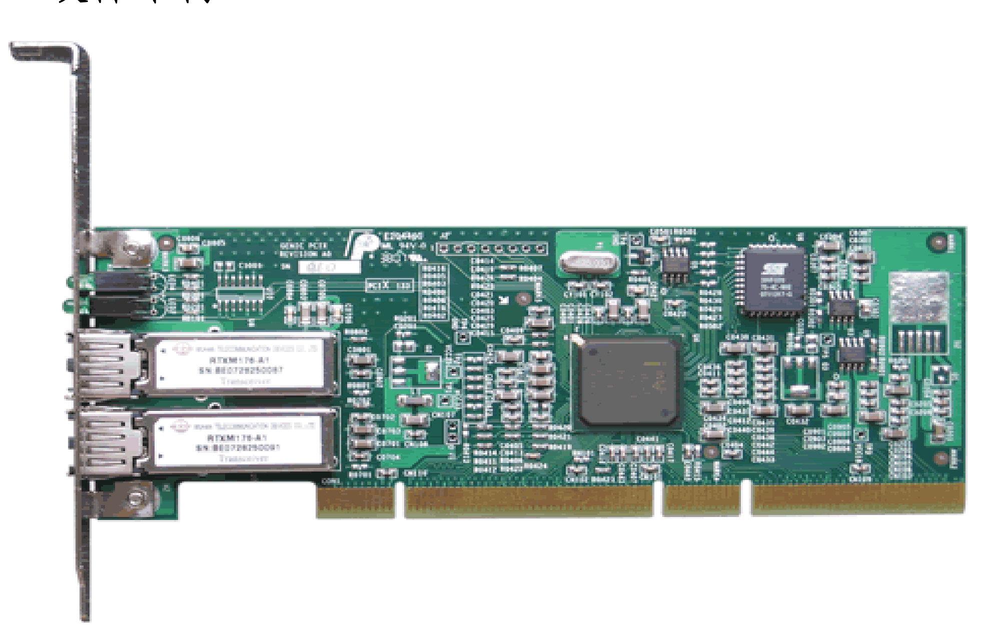


图 2-1 FS1002-X 正面结构图



图 2-2 FS1002-X 测面图

正面向上，光口从左至右为 2 口和 1 口，面板指示灯分为 2 列，分别对应 2 口和 1 口，每列上方为收发包指示灯。

2.1.2 系统配置清单

项 目	部件名称	数 量
1	千兆以太网卡	1 块
2	安装光盘、用户手册	1 套
3	1U 机箱挡板条配件	1 副

3 驱动安装

3.1 Windows 下驱动安装

本章介绍了 FS1000 系列千兆以太网卡在 windows 下驱动的安装方法。驱动安装文件为光盘 WINDOWS 目录下 FS1000_driver.exe。

3.1.1 驱动安装

1. 将千兆以太网卡插入系统中，双击 FS1000_driver.exe，出现如图 3-1 所示画面，点击“下一步”。

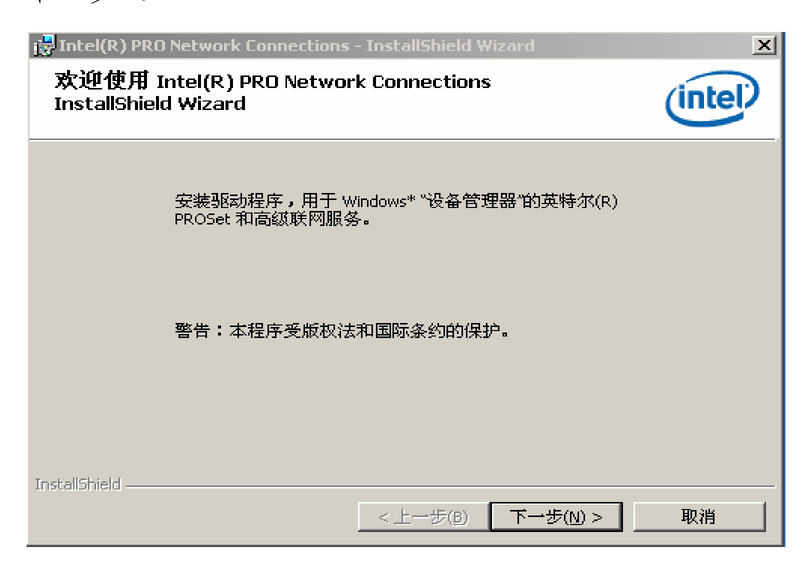


图 3-1 安装千兆以太网卡

2. 出现如图 3-2 所示画面，选择“我接受该许可协议中的条款”，然后点击“下一步”。

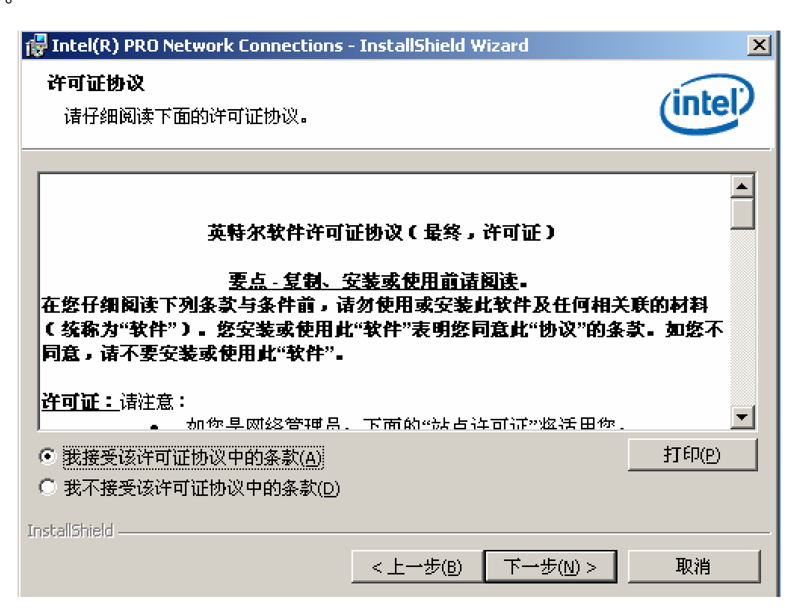


图 3-2 接受许可协议

3. 出现如图 3-3 所示画面，选择“驱动程序、英特尔 ROSet 和高级网络服务”，点击“下一步”。

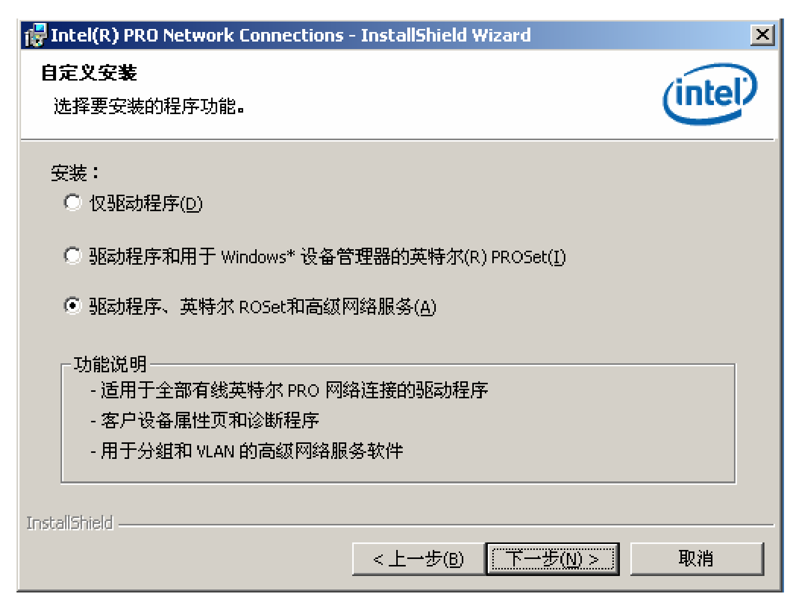


图 3-3 选择安装项目

4. 出现如图 3-4 所示画面，点击“安装”，开始安装千兆网卡驱动。



图 3-4 开始安装

5. 出现如图 3-5 所示画面，点击“下一步”。



图 3-5 安装千兆以太网卡驱动

6. 出现如图 3-6 所示画面，点击“完成”，此时表示已将驱动程序成功安装到系统中。

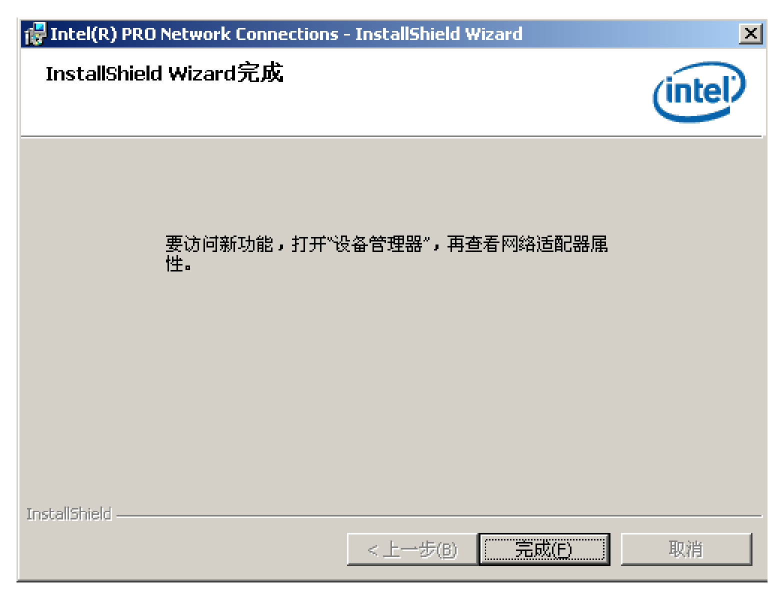


图 3-6 完成驱动程序安装

3.2 Linux 下驱动安装

Linux 下驱动的安装文件为光盘 LINUX 目录下的 e1000-7.6.5.tar.gz。

3.2.1 驱动安装

- 运行命令 tar xvzf e1000-7.6.5.tar.gz 解开软件包。
- 运行命令 cd e1000-7.6.5/src 进入 src 目录。
- 运行命令 make; make install 安装千兆以太网卡的驱动。

3.2.2 千兆以太网卡测试

安装好千兆以太网卡的驱动后，根据系统的不同，该千兆以太网卡的名字会有所变化。以本系统为例，本系统中该千兆以太网卡的名字为 eth2。运行命令 ifconfig eth2 up 启动千兆以太网卡，运行 tcpdump –i eth2 –nne，采集所有数据包。

4　应用编程接口

本章主要介绍了如何通过 Libpcap（Linux 平台下）或者 Winpcap（Windows 平台下）提供的接口来对千兆以太网卡进行收包操作。

4.1　接口概述

下表列出各调用接口的名称以及功能的概述,在其后部分将对每一接口及其功能进行详细的描述:

函数名	功能概述
pcap_open_live	获取采集句柄。
pcap_compile	解析规则和配置规则。
pcap_loop	开始进行采集.该函数内部处于循环状态。
pcap_breakloop	中断采集。
pcap_close	销毁采集句柄，释放资源。
pcap_stats	获取应用进程收包和丢包的统计信息。
pcap_version	库版本号。

4.2　基本数据结构

4.2.1　pcap_t

pcap_t 为设备句柄，用户使用该句柄访问设备和采集数据包。

4.2.2　pcap_handler

pcap_handler 为回调函数的指针，对采集到的数据包进行处理。该类型的声明如下:

```
typedef void (*pcap_handler) (u_char *usr_data, const struct pcap_pkthdr *hdr,    const u_char *p);
```

其中，usr_data 为用户自定义数据；hdr 为数据包信息，p 为数据包指针，后两项是系统传入的参数，用户直接使用即可。

4.2.3　pcap_pkthdr

pcap_pkthdr 包含数据包信息，目前只支持数据封包的长度。

```
struct pcap_pkthdr {
        struct timeval ts;      /* time stamp */
        bpf_u_int32 caplen;/* length of portion present */
        bpf_u_int32 len;        /* length this packet (off wire) */
    };
```

4.2.4 pcap_stat

pcap_stat 包含应用进程收包和丢包的统计信息，具体结构如下：

```
struct pcap_stat {
    u_int ps_recv;      /* number of packets received */
    u_int ps_drop;      /* number of packets dropped */
    u_int ps_ifdrop;    /* drops by interface XXX not yet supported */
};
```

4.3 功能详述

4.3.1 pcap_open_live

通过该函数获取采集句柄．应用程序通过采集句柄访问设备。

函数原型

pcap_t * pcap_open_live(const char *device, int snaplen, int promisc, int to_ms, char *errbuf)

入口参数

device	用于描述要打开设备的名称，在 linux 下为 ethN，N 取大等于零的整数。
snaplen	采集数据封包的最大长度。即如果封包实际含有大于 snaplen 的长度，但也只采集该封包的前 snaplen 个字节。
promisc	打开混杂模式。
to_ms	超时参数。

出口参数

errbuf	如果该参数不为 NULL，则 errbuf 指向的内存空间一定能够容纳 GCAP_ERROR_BUFFER_SIZE 个字符，使用该口进行过滤规则过程中发生错误时，该接口会将发生错误的描述信息放入该指针指向空间，如果该参数为 NULL，则表明用户并不在意错误描述信息。

返回值

创建成功，返回设备句柄，否则返回 NULL，如果 errbuf 不为 NULL，则通过 errbuf 传递错误信息。

4.3.2 pcap_compile

该函数用于解析规则和配置规则。

函数原型

int pcap_compile (pcap_t *handle, struct bpf_program *program, char *buf, int optimize,

bpf_u_int32 mask)

入口参数

handle	通过 pcap_open_live 获得的采集句柄.用该句柄访问设备。该参数无默认值。
program	一个 bpf_program 结构指针。
buf	保存需要设置的过滤规则。规则的关键字用空格分开，比如 buf = "tcp port 80" 此规则解析为过滤 tcp 类型的数据包并且数据包的目的或者源端口为 80。
optimize	指定结果代码的优化程度。
mask	本地网络的网络掩码。

返回值

成功返回 0，否则返回-1，错误原因可能是用户向该函数传递了非法的参数。

4.3.3 pcap_loop

开始进行采集.该函数内部处于循环状态，直到用户规定的 cnt 个数据封包为止.该函数每接收到一个数据封包，都会调用 callback 注册函数对数据包处理。用户可调用下面描述的 gcap_breakloop 结束循环状态。

函数原型

int pcap_loop (pcap_t *handle, int cnt, pcap_handler callback, u_char *user)

入口参数

handle	通过 pcap_open_live 获得的设备句柄.用句柄访问设备。该参数无默认值。
cnt	限定接收数据包的个数，当接收满 cnt 个数据后 pcap_loop 返回。当 cnt 小于等于零时表示无限循环接收收据包。当没有数据包时此函数返回。
callback	注册的回调函数，该函数每接收到一个数据封包，都会调用 callback 注册函数对数据包处理。
user	用户自定义数据，传递给 callback 使用。

返回值

成功返回 0，否则返回-1，错误原因可能是用户传递了错误的参数。

4.3.4 pcap_breakloop

停止在指定数据采集句柄上的所有数据采集处理。如果在一线程中对某采集句柄进行了 pcap_loop 操作，可以在另外的线程中调用 pcap_breakloop 来中止该循环。

函数原型

void pcap_breakloop (pcap_t *handle)

入口参数

handle	通过 pcap_open_live 获得的设备句柄.用句柄访问设备。该参数无默认值。

4.3.5 pcap_close

销毁采集句柄.采集工作完成后,使用该函数释放采集句柄本身所使用的资源。

函数原型

int pcap_close (pcap_t * handle)

入口参数

handle	待销毁的采集句柄。

返回值

销毁成功返回 0，否则返回-1，失败的原因可能是该句柄仍然处于使用中，或者是用户向该函数传递了非法的参数。

4.3.6 pcap_stats

获取对应采集描述句柄的采集信息.

函数原型

int pcap_stats (pcap_t *handle, struct pcap_stat *stats)

入口参数

handle	通过 pcap_open_live 获得的采集句柄。用句柄访问设备。该参数无默认值。

出口参数

stats	pcap 对应的采集句柄的统计信息。包括接收到和丢弃的数据包统计信息。

返回值

销毁成功返回 0，否则返回-1。

4.3.7 pcap_version

获取 libpcap 库的版本号。

函数原型

uint32_t pcap_version()

返回值

libpcap 库的版本号，高 16 位返回主版本号，低 16 位返回次版本号。

4.4 接口函数使用举例

4.4.1 举例一 接收 tcp port 80 数据包

```
#include <pcap.h>
#include <stdio.h>

void pcap_print(u_char *user, const struct pcap_pkthdr *h, const u_char
*p)
{
    int i;

    for ( i = 0; i <h->len; i++)
    {
            printf( "%02x-", p[i] );
    }
    printf("\n");
}


int main()
{
   pcap_t *handle; /* the handle */
   char *dev= “eth0” ; /* the device */
   char errbuf[PCAP_ERRBUF_SIZE]; /* err buf */
   struct bpf_program filter; /* filter, no used */
   char filter_app[] = "tcp port 80"; /* the rule */
   bpf_u_int32 mask; /* no used */
   bpf_u_int32 net; /* no used */
   struct pcap_pkthdr header; /* header pointer */
   const u_char *packet;

   /* Define the device */
   handle = pcap_open_live(dev, 1600, 1, 0, errbuf);
   pcap_compile(handle, &filter, filter_app, 0, net);
   pcap_setfilter(handle, &filter);
    pcap_loop(handle, -1, pcap_print, NULL); //this function
                                 //to test the pcap_loop
   pcap_close(handle);
   return(0);
}
```



产品合格证

检验员：______________ 日期：________________